

Sector Logic Board

The ATLAS Muon Trigger Sector Logic/RX Data Acquisition Board

Author: Federico Spila

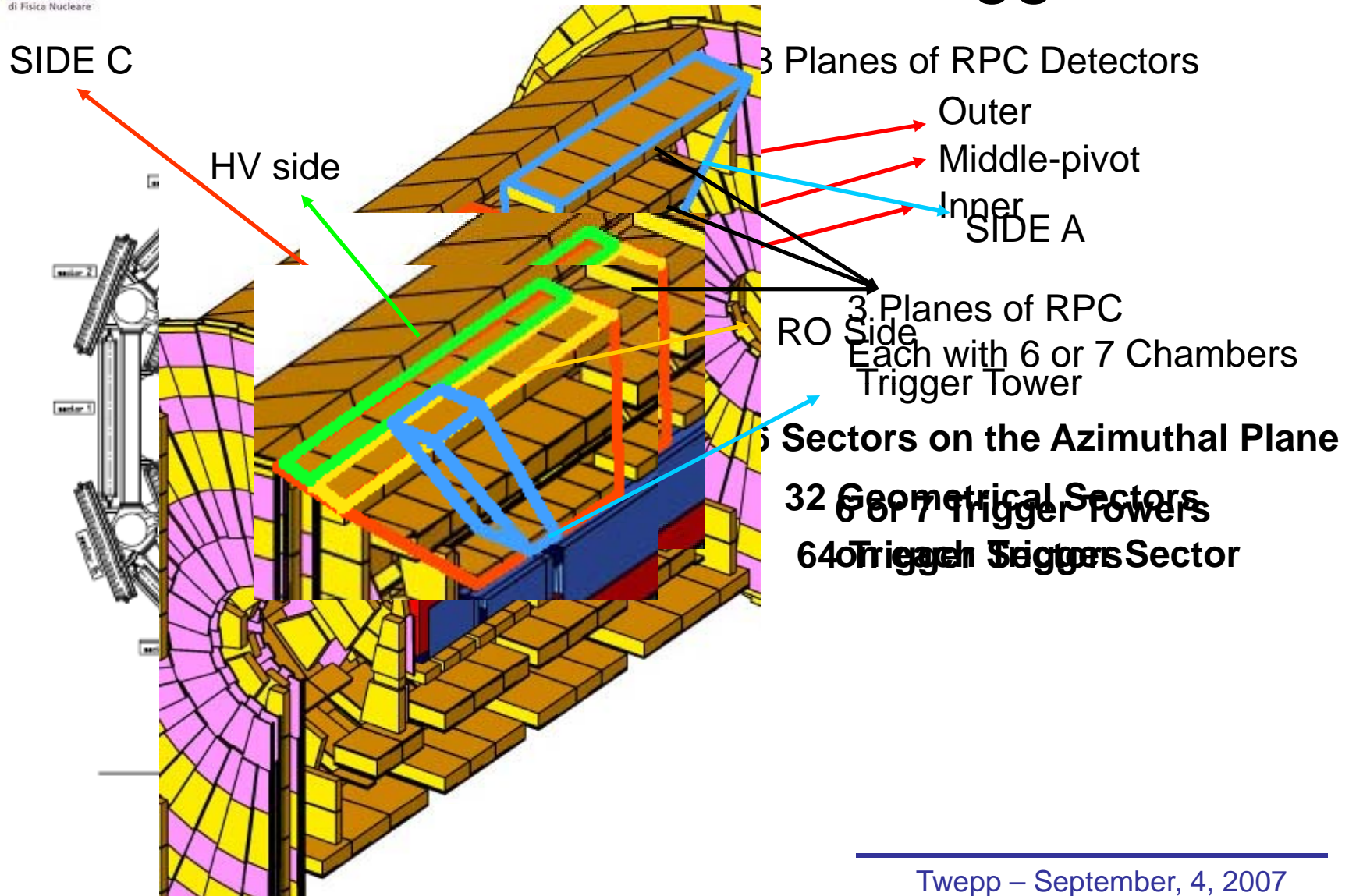


INFN Rome

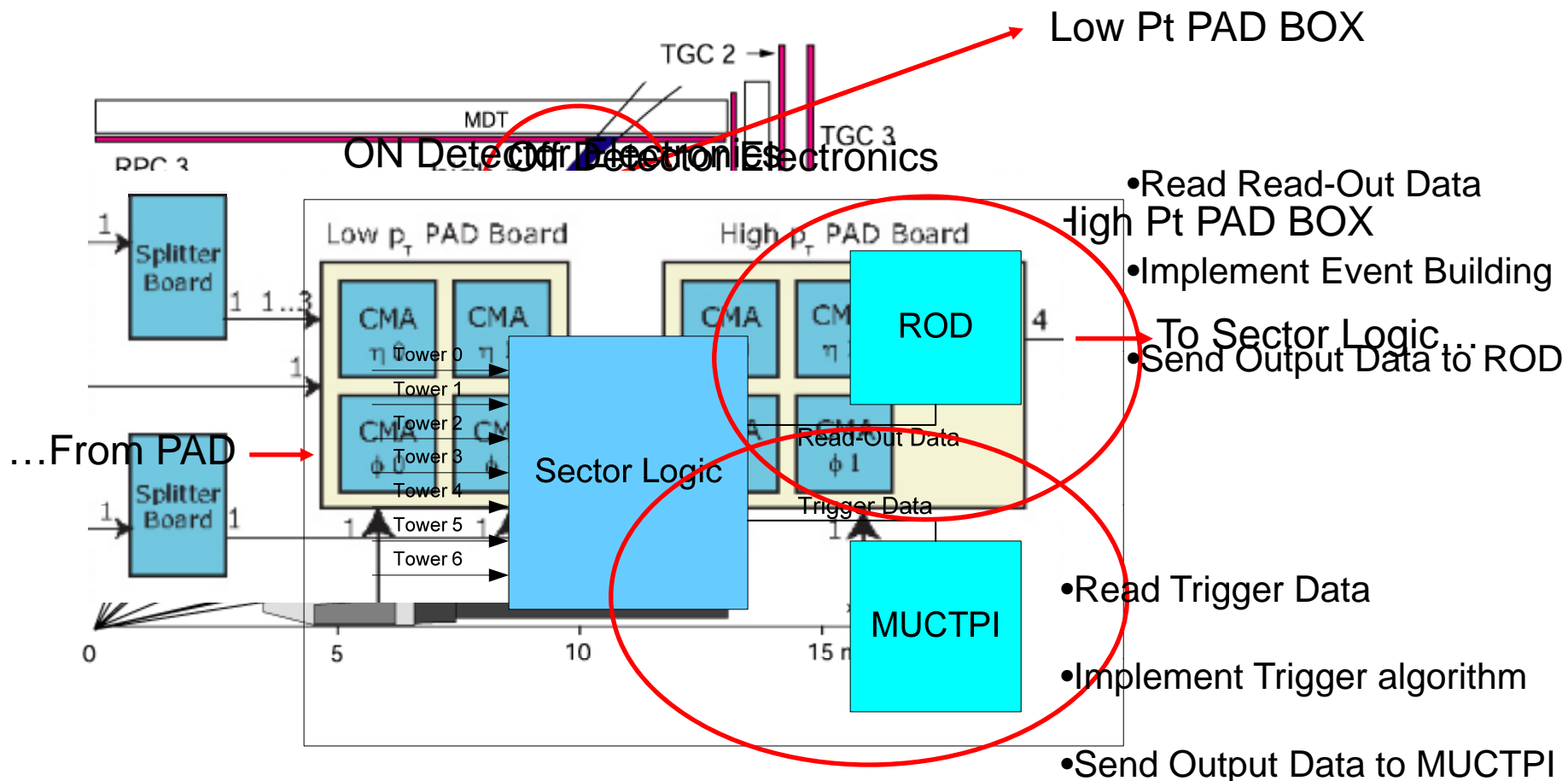


Twepp – September, 4, 2007

1/13 LVL1 Muon Barrel Trigger



2/13 Lvl1 Trigger Electronics



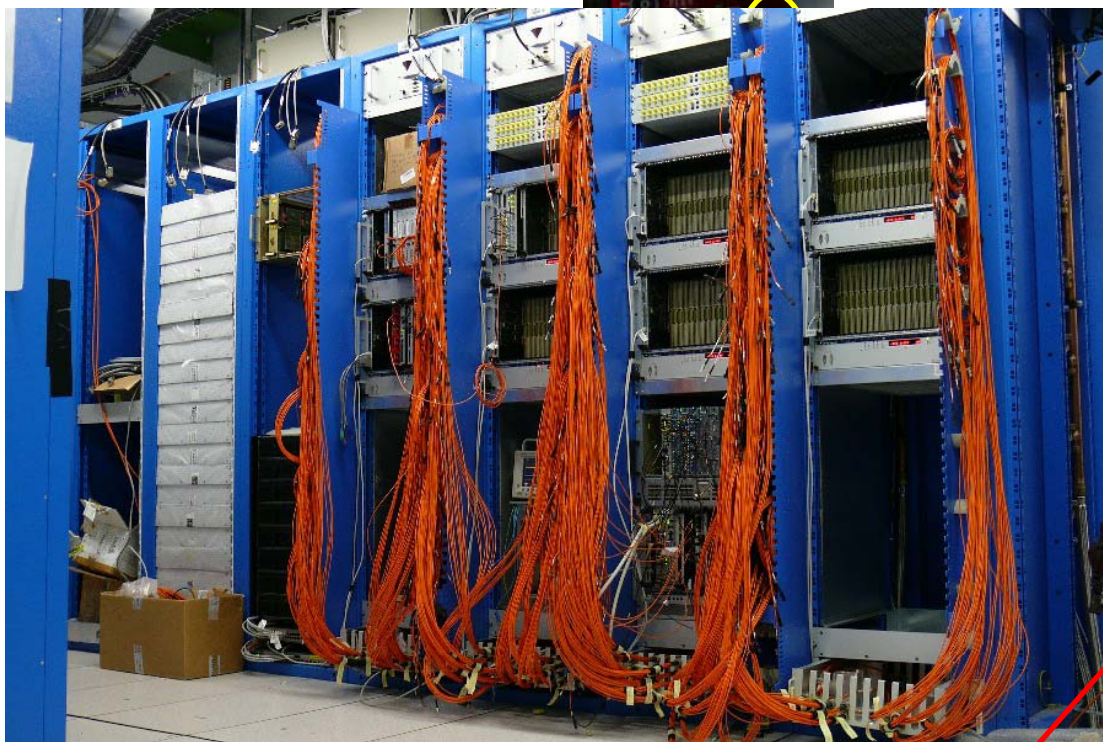
3/13 VME Crates

USA 15 Room 16 VME Crates

MUCTPI-Interface

Sector Logic

Head-Out Data: Backplane Connection



Each takes data from
2 Geometrical Sectors

Each takes data from
2 Trigger Sectors

Central Trigger
Processor

In Total 64 Sector Logic
Boards

MUCTPI
receivers

External

L1 Barrel Trigger
DS BUS

L1 End-Cap Trigger
...To MUCTPI board

4/13 ROD-BUS

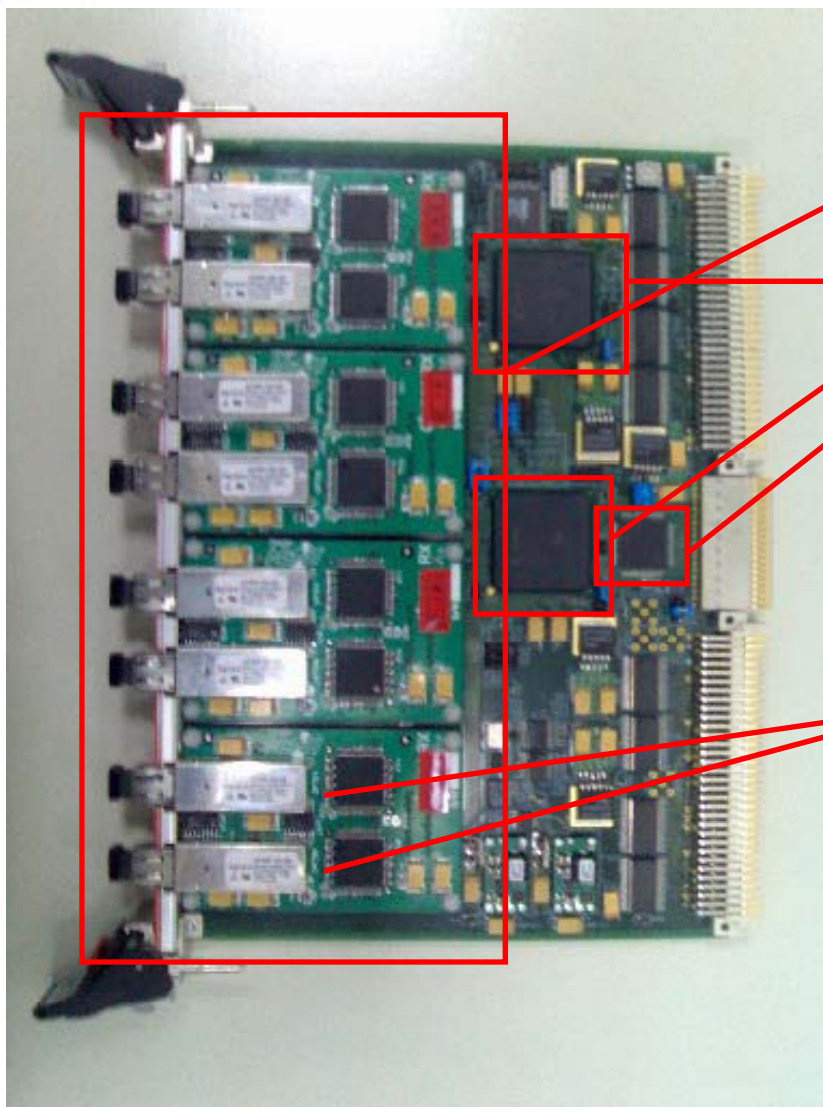


SL <-> ROD
Real ROD Block

- Sends Read-Out Data

1. MUCTPI-Interface
 2. ROD BUS mounted on a VME Back-Panel
 3. ROD reads 20 Sector Logics
 4. Sector Logic <-> MUCTPI-Interface
 5. MUCTPI-Interface
- Receives JTC Signals
 - Receives Service Signals

5/13 Sector Logic Schema



4 G2Link RX Cards

Xilinx Virtex2 XC2V 1000
Xilinx Virtex2 XC2V 2000
Serializer chip

VME Communication

JTAG

40-bit SerDes logic features

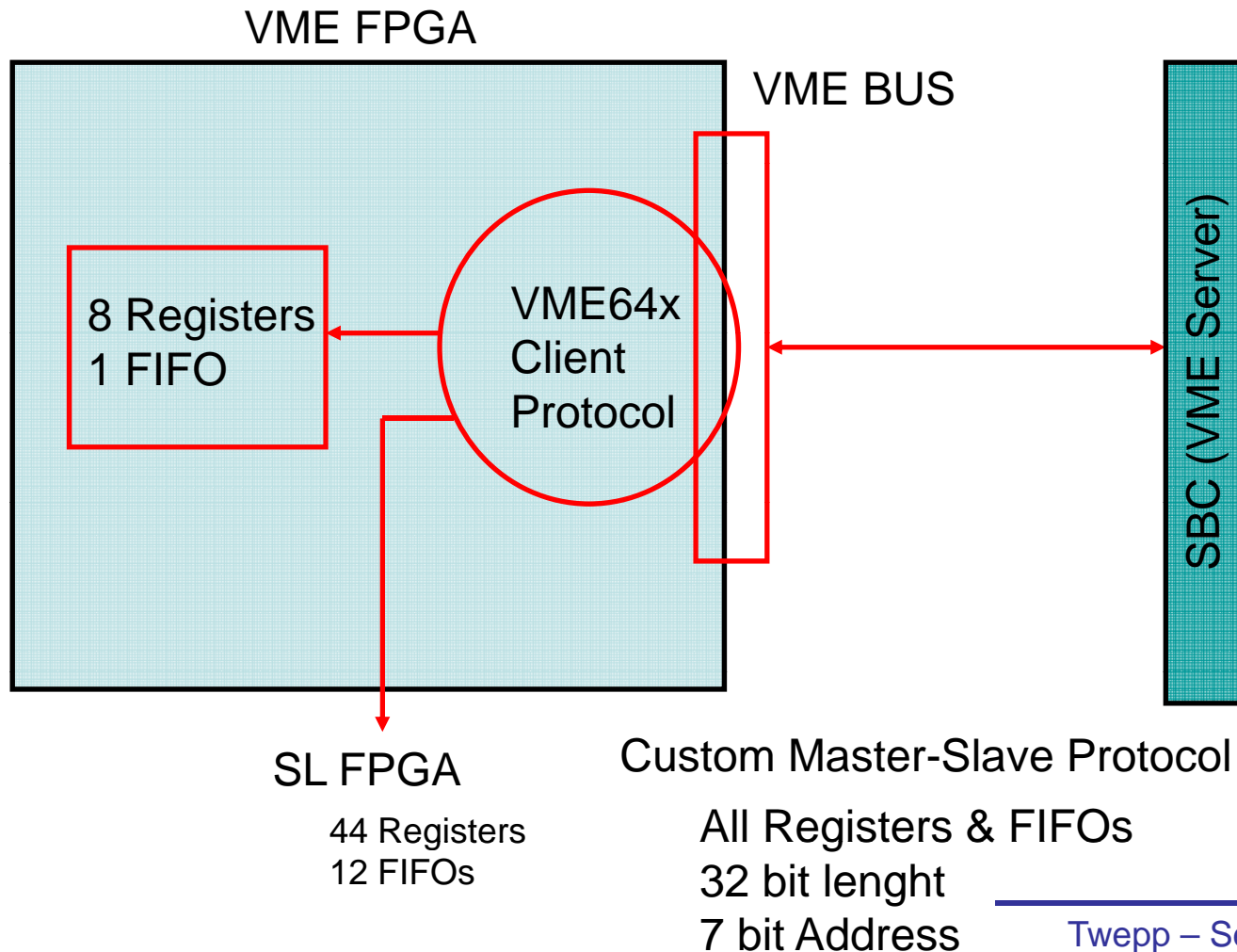
->

8 bit LVDS output stream

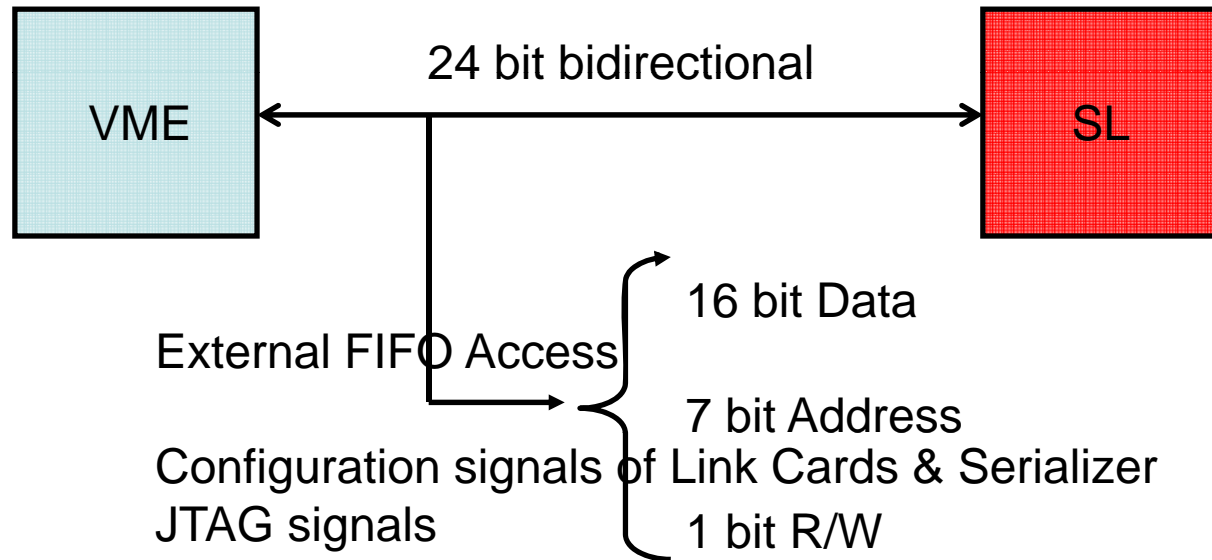
Each with 2 Optical link receivers
Speedrate 240 Mbyte/s

VME FPGA
SL FPGA

6/13 VME FPGA Firmware



7/13 Master-Slave Protocol

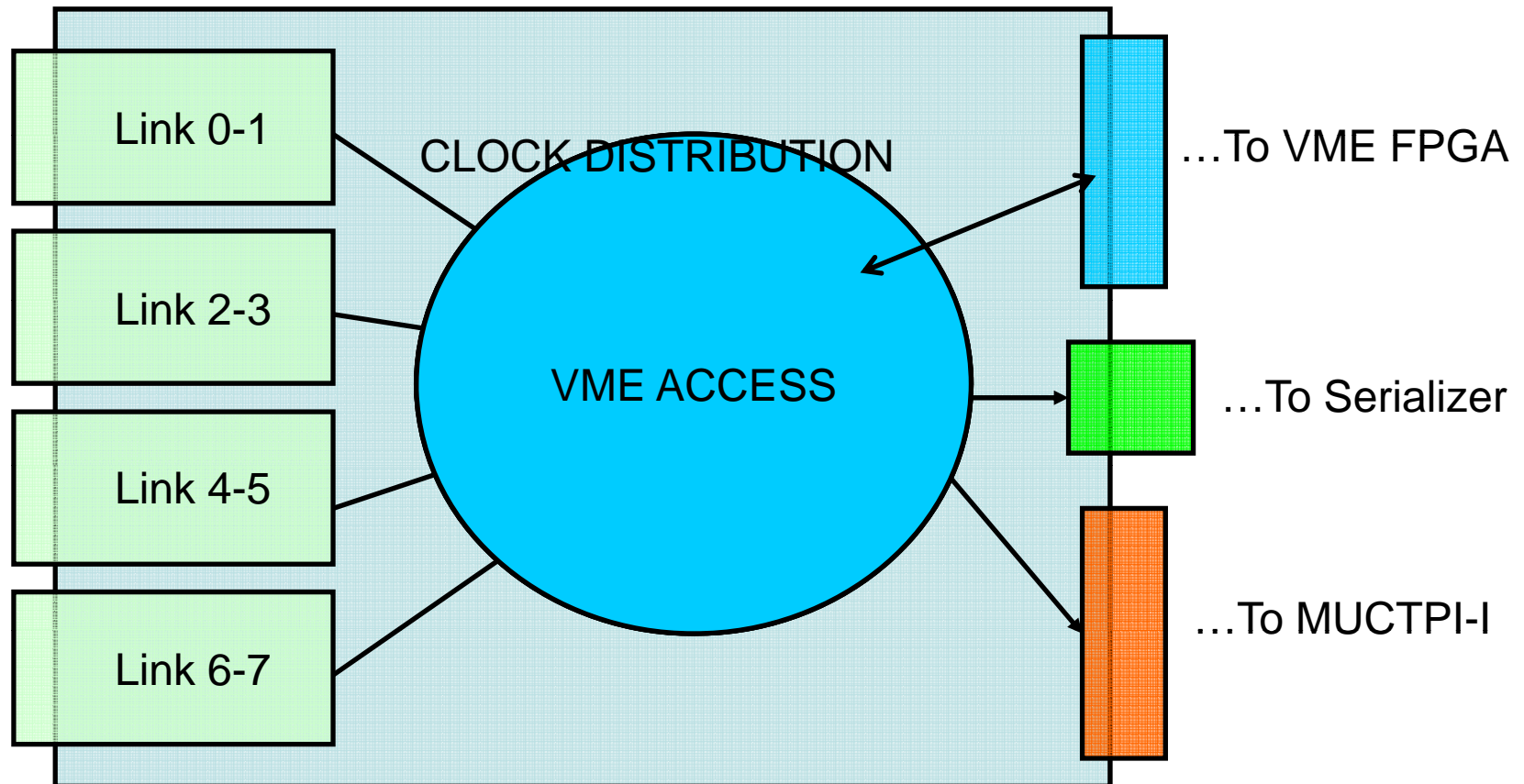


VME FPGA is the Master

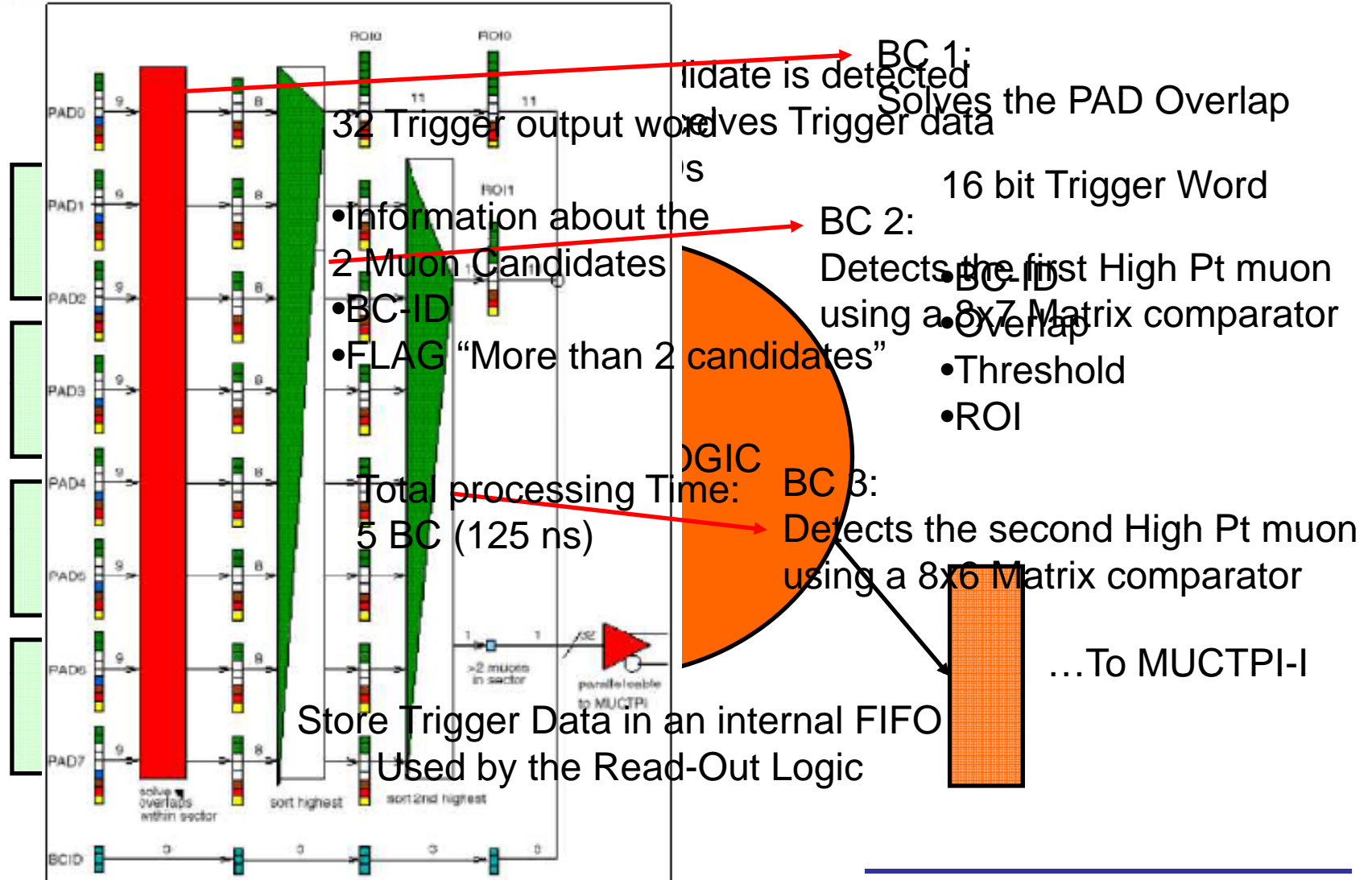
Transmit first the low significant 2 bytes,
after the more significant 2 bytes

8/13 SL FPGA Firmware

SL FPGA

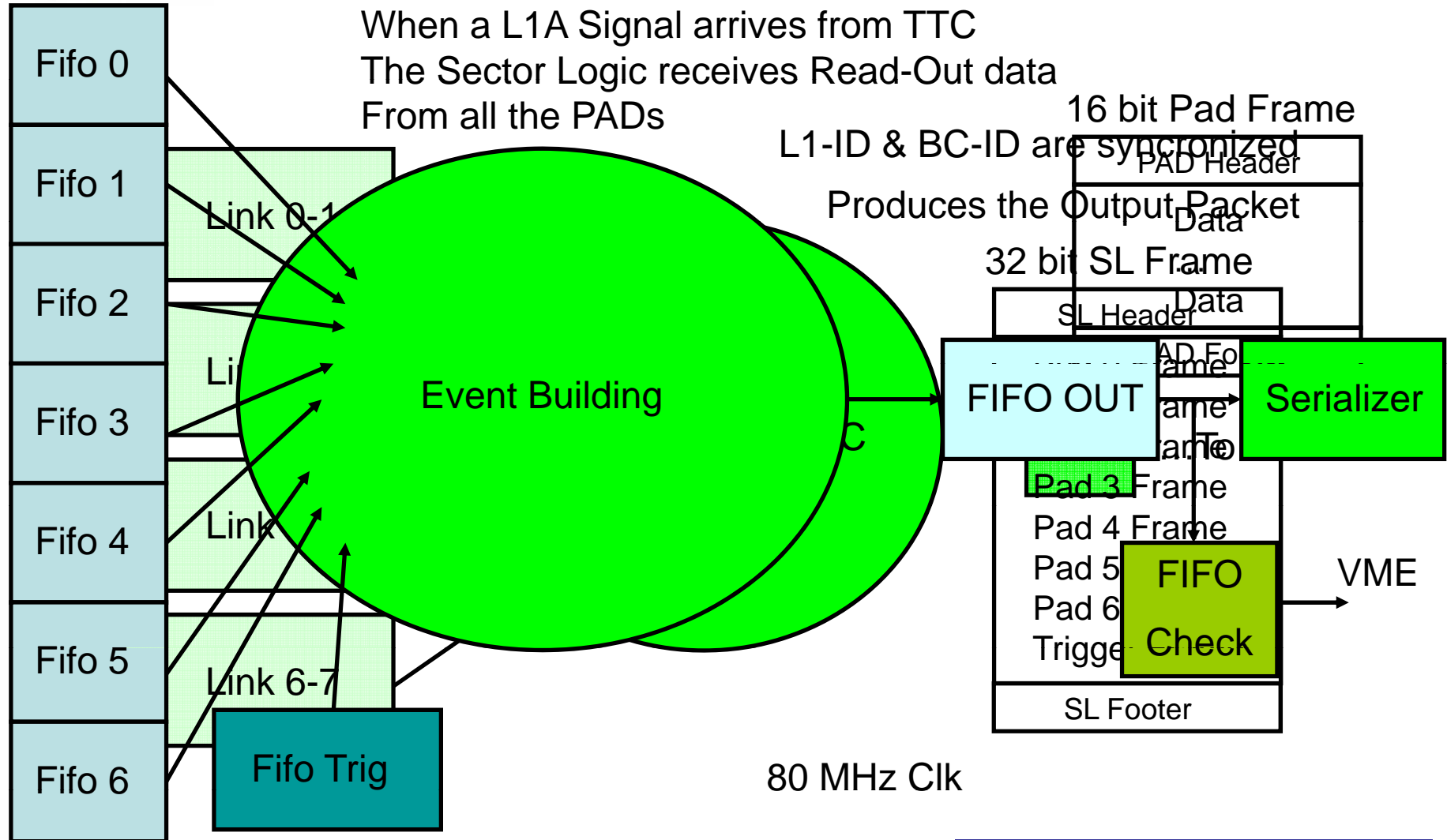


9/13 Trigger Logic



10/13 Read-Out Logic

When a L1A Signal arrives from TTC
 The Sector Logic receives Read-Out data
 From all the PADS



11/13 VME SL Logic

GENERAL PURPOSE BOARD for Testing

- PADs
- Sector Logics
- RODs
- MUCTPIs

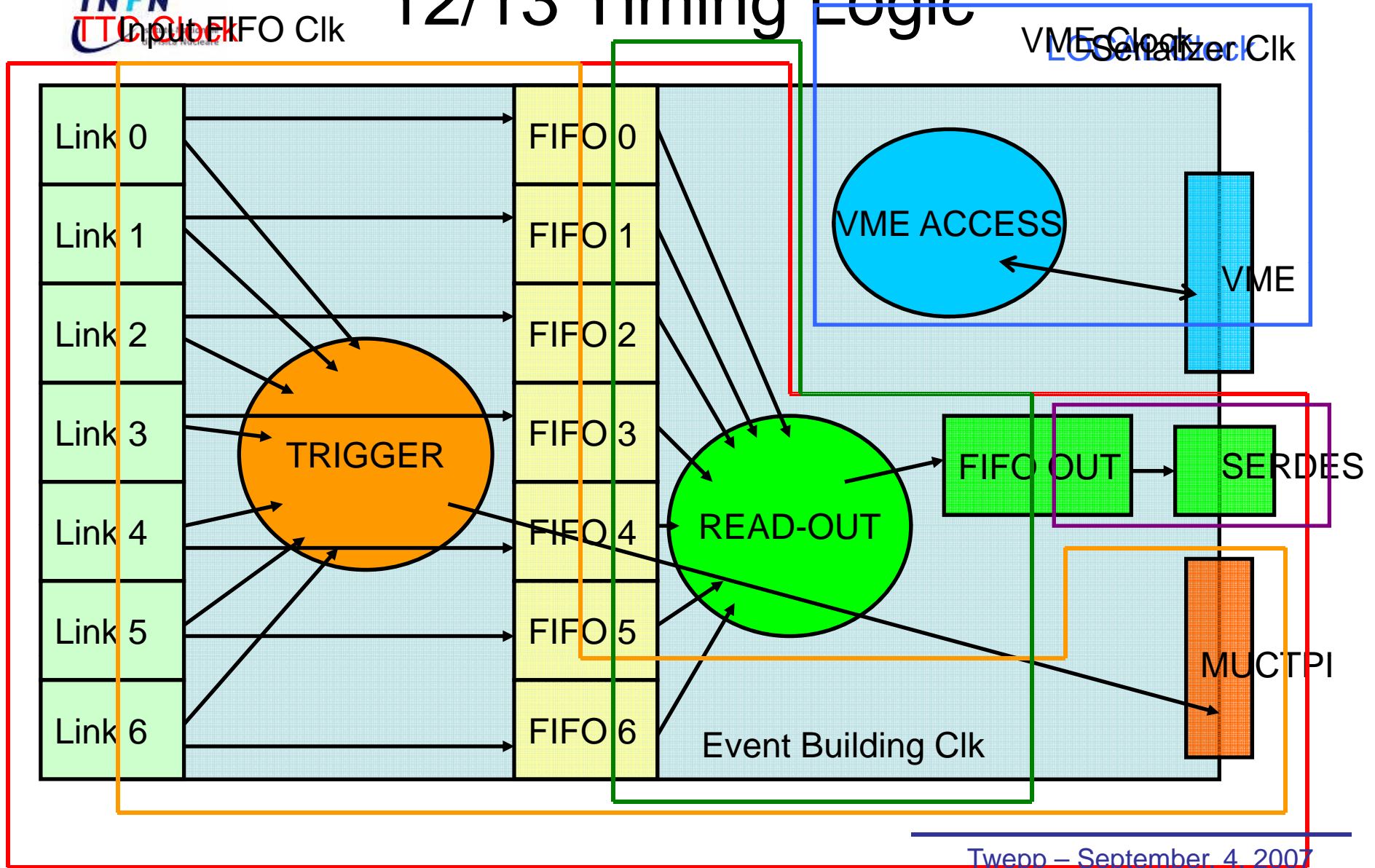
...To VME FPGA

VME ACCESS

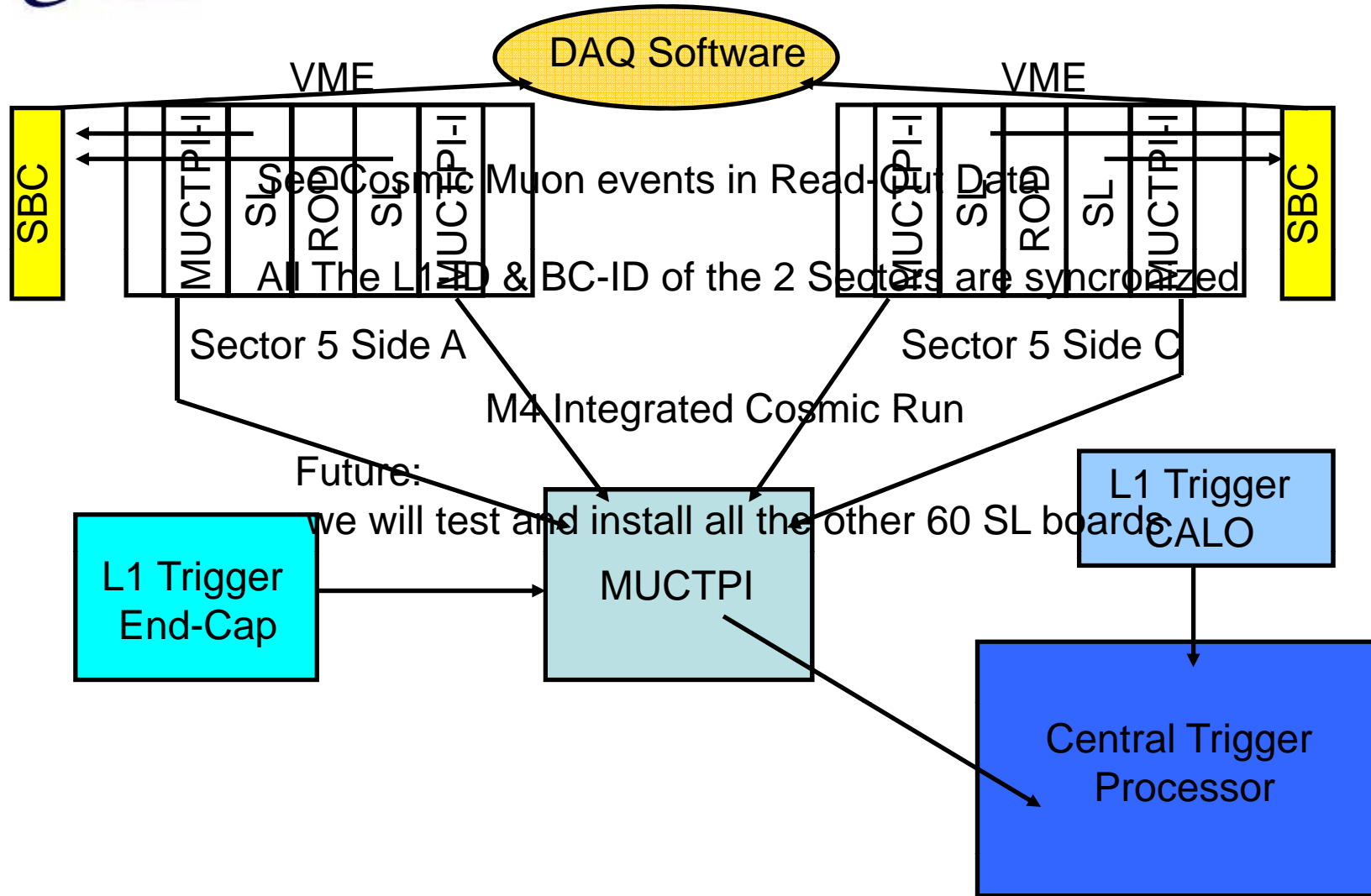
Master-Slave Custom Protocol

- Write input FIFOs simulating PADs
- Write Timing signals simulating the TTC
- Write the Output Data directly to ROD
- Write the Output Data directly to MUCTPI
- Mounting G2Link TX, we can emulate the PAD

12/13 Timing Logic



13/13 Conclusions & M4 RUN



THE END